

LOW-COMPLEXITY ASYMMETRIC MULTIVIEW VIDEO CODING

Ying Chen¹, Shujie Liu², Ye-Kui Wang³, Miska M. Hannuksela³, Houqiang Li², Moncef Gabbouj¹

¹Department of Signal Processing, Tampere University of Technology

²University of Science and Technology of China

³Nokia Research Center

ABSTRACT

Multiview video coding (MVC) is currently under development by the Joint Video Team (JVT) as an extension to Advanced Video Coding (H264/AVC). Based on the suppression theory in binocular vision, the fidelity of one of the two views of a stereoscopic display can be reduced without noticeable degradation of subjective quality. Thus, in MVC, a subset of views can be coded with lower spatial resolution at negligible cost to subjective quality. Due to different resolutions, a downsampling process is required in an MVC decoder in order to enable motion compensation (MC) between views. In this paper, a low-complexity MC algorithm is proposed for MVC to enable inter-view prediction between pictures with different resolutions. It requires lower memory consumption and lower computational complexity compared with the conventional downsampled inter-view prediction, while providing comparable efficiency, as shown by the simulation results.

Index Terms— Multiview video coding, asymmetric coding, motion compensation, decoded picture buffer

1. INTRODUCTION

Multiview video technologies have gained increasing interest recently. In multiview applications, the original video content is a group of video sequences captured by multiple cameras at the same time from the same scene. As the views have high correlation between each other, the JVT has worked to reduce the inter-view redundancy to obtain improved coding efficiency for the MVC standard [1], which will become an extension to H.264/AVC. Many of the display arrangements for multiview video are based on rendering of a different image to viewer's left and right eye. Hence, only two views are observed at a time in many typical MVC applications, such as 3D TV [2]. Based on the concept of asymmetric coding, one view in a stereoscopic pair can be coded with lower quality, while the perceptual quality degradation for the stereoscopic display is not noticeable by human eyes in comparison to the case when both of them are coded with equally high quality [3].

Therefore, a subset of all the views can be coded in lower resolution, e.g., quarter resolution, compared to the others and upsampled when being displayed. This scenario, referred to as asymmetric multiview video coding, requires less transmission bandwidth as well as lower complexity at the decoder for those low-resolution views.

For stereoscopic video, an approach has been proposed to code one view with high resolution while the other with half or quarter resolution [4]. In addition, there have been contributions to the JVT about asymmetric coding [5][6]. In the schemes proposed in [4] and [5], a part of views (e.g., every other view) is coded in the original resolution, while the remaining views can be coded with quarter resolution. To enable inter-view prediction between pictures with different resolutions, [4] and [5] took the following approach. The high-resolution pictures are downsampled before used for motion compensation (MC) by low-resolution pictures. Due to the fact that a high-resolution decoded picture is used as an inter prediction reference picture when the pictures in the same view (hence with the same resolution) are coded, the original version (with high resolution) is also kept in the decoded picture buffer (DPB). Since both the downsampled decoded picture and the full-resolution decoded picture coexist for one coded picture, the DPB size is inevitably increased. Instead of storing both downsampled and full-resolution decoded pictures, on-the-fly downsampling could be applied, but it would increase the computational demands for real-time processing. Therefore, the conventional solutions require either more memory or higher computational complexity.

In this paper, we propose a scheme with low complexity MC for the asymmetric MVC. This algorithm reduces the complexity for the decoding of the low-resolution views without increasing DPB size. Simulation results show that the proposed scheme provides comparable efficiency with the conventional solution.

This paper is organized as follows. In Section 2 and 3, asymmetric MVC and MC in H.264/AVC are reviewed. The proposed method is introduced in section 4. In Section 5, simulation results for coding efficiency of the proposed method are given, and then complexity reduction of the method is analyzed. Section 6 concludes the paper.

2. ASYMMETRIC MULTIVIEW VIDEO CODING

2.1. Inter-view prediction in MVC

A typical prediction structure of MVC is shown in Fig. 1, wherein T stands for the time axis and S stands for the view axis. Pictures in each view form a hierarchical bi-predictive (B) temporal prediction structure. The base view (view 0) is independently coded. For a picture in other views, inter-view prediction can be applied. Inter-view prediction is realized by placing the inter-view reference picture (inter-view picture for simplicity) into the reference picture lists and then the inter-view picture is utilized similarly to an inter prediction reference picture, and the inter-view prediction process is similar to the normal inter prediction process in the temporal direction. As shown in Fig. 1, a picture can use pictures in other views within the same time instance for inter-view prediction. In the joint draft of MVC [1], the inter-view prediction relationship is indicated as view dependency in the sequence parameter set MVC extension, wherein dependent views are signaled separately for the views that may be used as reference pictures in the two reference picture lists, namely list 0 and list 1. The dependent views corresponding to list 0/list 1 are also called forward/backward dependent views. A view that has both forward and backward dependent views is called a “B-view”. For example, views 1, 3, 5 in Fig. 1 are “B-views”.

In MVC, an anchor picture is a picture in a view that can be correctly decoded without the decoding of any earlier access unit in decoding order (i.e. bitstream order). Other pictures are non-anchor pictures.

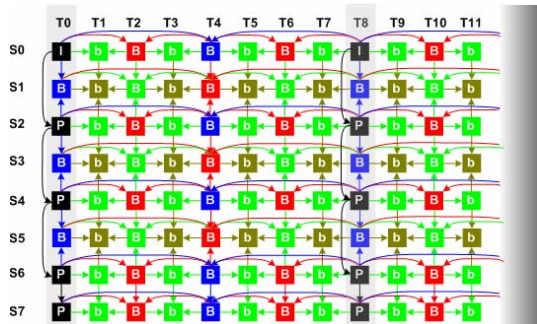


Fig. 1: Typical MVC prediction structure.

2.2. Asymmetric multiview video coding

In asymmetric MVC, for the stereoscopic scenario, two views are coded, one is in a higher resolution (e.g. VGA) and the other is in a lower resolution (e.g. QVGA). Downsampling is required to support the inter-view prediction between these two views. After that, MC of H.264/AVC can be applied. This asymmetric MVC approach is motivated by suppression theory of binocular vision [3], which indicates that the perceived sharpness and

depth effect of a mixed-resolution stereoscopic pair is dominated by the higher-quality component, which can correspond to the right-eye, for example [4]. A 2D video communication system then can be enhanced to a 3D video communication system based on stereoscopic display, with acceptable transmission bandwidth increase, e.g., in a DVB-H (Digital Video Broadcasting –Handheld) system [7], and with reasonable decoder complexity increase, e.g. around 25%, compared with existing H.264/AVC decoders.

A picture in a DPB can play two different roles: an inter prediction reference picture for the following pictures in the same view, and an inter-view picture for the pictures in the same time instance. In asymmetric MVC, an inter-view picture, when referenced by a picture with a low resolution, needs to be downsampled to apply the conventional MC. Therefore, either the downsampled picture must be stored in the decoded picture buffer (DPB) or there must be an on-the-fly downsampling process. If a downsampled picture is added into the DPB, the required DPB size increases. If on-the-fly downsampling is used, the complexity increases, especially when the picture is used frequently as inter-view picture by lower-resolution pictures. These problems get worse when typical downsampling filter such as the MPEG-4 downsampling filter is applied. The MPEG-4 downsampling filter is $[2, 0, -4, -3, 5, 19, 26, 19, 5, -3, -4, 0, 2]/64$, which is a 13-tap filter and has 11 non-zero taps.

3. MOTION COMPENSATION IN H.264/AVC

In H.264/AVC, the accuracy of MC is in units of one quarter of the distance between luma samples. In case the motion vector points to an integer-sample position, the prediction signal consists of the corresponding samples of the reference picture; otherwise the prediction signal is obtained using interpolation to generate sample values at non-integer sample positions. The prediction values at half-sample positions are obtained by applying a one-dimensional 6-tap filter first horizontally and then vertically. The 6-tap filter utilized in H.264/AVC half-sample interpolation is $[1, -5, 20, 20, -5, 1]/32$ (named AVC filter for simplicity). Prediction values at quarter-sample positions are generated by averaging samples at integer-sample and half-sample positions. In the chroma component, a motion vector can point to integer sample, half-sample, quarter-samples or 1/8-sample positions. When the motion vector points to a non-integer sample position in a chroma component, the prediction values for chroma are obtained by utilizing the bilinear filter.

4. PROPOSED LOW-COMPLEXITY ASYMMETRIC MVC

In H.264/AVC, when a motion vector points to non-integer-sample positions, an area of the reference picture needs to

be interpolated. However, in the asymmetric MVC, since the inter-view picture is already with double width and height as the picture being coded, MC can be done without interpolation for the half-sample values. The downsampling of the inter-view picture to the same resolution, as well as the interpolation of values at half-sample positions can be avoided. Based on this, we provide a low-complexity MC approach for the asymmetric MVC. The proposed MC process consists of the following sub-processes.

4.1. Motion vector scaling

The motion vector to be used in the proposed MC process is first scaled as follows: $mv' = 2mv$, wherein mv is the original motion vector and mv' is the scaled motion vector.

4.2. Luma motion compensation

In our proposed method, a scaled motion vector points to even-sample positions (when the original motion vector points to integer-sample positions), odd-sample positions (when the original motion vector points to half-sample positions), or half-sample positions (when the original motion vector points to quarter-sample positions) in the inter-view reference picture.

In Fig. 2, upper-case letters indicate samples on the full-sample grid, while lower case letters indicate samples in between at full-samples. When the scaled motion vector points to an odd sample position or an even sample positions, no interpolation is required and the sample values in those positions can be directly used. Similar to the method used in H.264/AVC for generating the luma values for quarter-sample positions, when the scaled motion vector points to a half-sample position, we average the integer samples.

4.3. Chroma motion compensation

Chroma MC is carried out in the same way as in H.264/AVC by applying the scaled motion vector to the chroma components of the inter-view reference picture. The scaled motion vector points to integer-sample positions, half-sample positions, or quarter-sample positions, but never points to 1/8 sample positions. As in H.264/AVC, the bilinear filter as follows and also shown in Fig. 3 is still used for interpolation of non-integer positions.

$$v = ((s - d_x)(s - d_y)A + d_x(s - d_y)B + (s - d_x)d_yC + d_xd_yD + s^2/2) / s^2$$

wherein A, B, C, D are the values of the integer samples and v is the value of the interpolated non-integer sample, s is 4 and d_x and d_y can be only 1, 2 or 3, while in H.264/AVC, s is 8 and d_x and d_y can be a value from 1 to 7.

5. SIMULATION AND ANALYSIS

5.1. Performance of the proposed approach

The proposed MC algorithm was implemented into the MVC reference software, JMVM (Joint Multiview Video Model) version 5. The low-resolution input views were generated by the MPEG-4 downsampling filter. The decoded low-resolution video was upsampled by the AVC filter for PSNR (luma peak signal-to-noise) calculation.

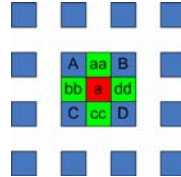


Fig. 2 Interpolation for half-sample luma values in the high-resolution picture.

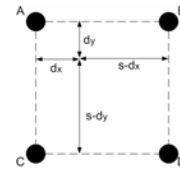


Fig. 3: Bilinear interpolation for chroma values in the non-integer positions.

Rate distortion (RD) performance was compared in two scenarios. The first scenario followed the JVT common test conditions [8] and the second one was the stereoscopic case, wherein two views were coded: view 0 was the base view and view 1 was of lower resolution and the pictures of view 1 are all depend on pictures (in the same time instance) in view 0. In both scenarios, three methods were compared: the proposed asymmetric MVC (PRO); the conventional asymmetric MVC (CON), wherein MPEG-4 downsampling was used [5]; and the simulcast MVC (SIM). When under common test conditions, SIM coded the base view and the “P-views” in one bit-stream with unchanged view dependencies and original spatial resolution. However, the “B-views” were coded into another MVC bitstream with quarter of the original resolution and with linear prediction structure. That is, e.g., if views 1, 3, 5 are the “B-views”, view 1 is the forward dependent view of view 3, and view 3 is the forward dependent view of view 5. In stereoscopic case, there was only one high-resolution view (base view: view 0) and one low-resolution view (view 1) and thus the simulcast MVC was actually simulcast H.264/AVC.

The tested sequences were: *Exit, Ballroom, Rena, Race1, Akko&Kayo, Breakdancers* and *Uli*.

The performance comparisons for PRO and CON as well as simulcast are listed in Table 1. The bit-rate and PSNR values are generated for all the views. The asymmetric MVC approaches outperform simulcast MVC, and PRO has almost the same efficiency as that of the CON. Note that, in the tables, a bit-rate saving or Δ PSNR greater than zero indicates that the left method is better than the right one. Results are generated using the Bjontegaard measurement [9]. The curves (representing the bit-rate and PSNR values among the views) for *Akko&Kayo* are shown in Fig. 4, to save space, the RD curves for the “B-views” (the three left curves) and the curves for all the views (the three right curves) are shown in the same figure.

With the comparison results (only for view 1) shown in Table 2, similar conclusion can be reached for stereoscopic

case. Table 1 and 2 also show that asymmetric MVC is an efficient tool compared to simulcast.

Table 1. Comparison of PRO to conventional CON and simulcast (SIM) under common test conditions (all views)

Sequence	PRO vs CON		PRO vs SIM	
	Bit-rate saving	Δ PSNR	Bit-rate saving	Δ PSNR
<i>Akko&Kayo</i>	1.61%	0.073	22.43%	0.923
<i>Ballroom</i>	1.65%	0.058	16.29%	0.510
<i>Exit</i>	1.44%	0.039	12.59%	0.304
<i>Race1</i>	1.66%	0.056	12.58%	0.419
<i>Rena</i>	-0.42%	-0.020	18.57%	0.746
<i>Breakdancers</i>	0.52%	0.011	14.40%	0.292
<i>Uli</i>	0.03%	0.001	0.97%	0.034
Average	0.93%	0.031	13.98%	0.451

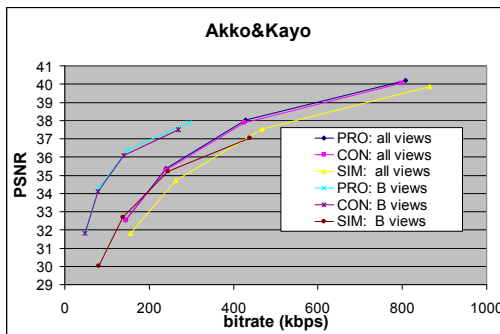


Fig. 4: RD curves for "Akko&Kayo".

Table 2. Comparison of PRO to CON method and SIM for stereoscopic video (view 1 only)

Sequence	PRO vs CON		PRO vs SIM	
	Bit-rate saving	Δ PSNR	Bit-rate saving	Δ PSNR
<i>Akko&Kayo</i>	2.49%	0.082	144.14%	3.339
<i>Ballroom</i>	-0.07%	0.003	75.06%	1.743
<i>Exit</i>	2.79%	0.067	59.48%	1.121
<i>Race1</i>	1.66%	0.056	12.58%	0.056
<i>Rena</i>	-4.05%	-0.134	126.29%	0.419
<i>Breakdancers</i>	-0.92%	-0.022	73.93%	2.821
<i>Uli</i>	-0.19%	-0.005	2.81%	0.079
Average	0.24%	0.007	70.61%	1.368

5.2. Complexity analysis

In H.264/AVC, interpolation of the half-sample values for luma is the major part of the entire interpolation process at the decoder, and takes around 40% of the decoder execution time [10]. The results in [10] are for the Baseline profile. For the Main profile or a High profile, interpolation will take a higher share of the decoding complexity because bi-predicted (B) pictures are present, especially when most of the pictures are B pictures, which is the case in hierarchical B prediction structure as shown in Fig. 1. These

interpolation computations are saved for anchor pictures in our approach. For non-anchor pictures, similarly, in average approximately 40% complexity reduction can be expected for macroblocks coded using inter-view prediction.

If on-the-fly downsampling is utilized, it requires even higher complexity than interpolation because the downsampling filter, e.g., the MPEG downsampling filter, normally has many non-zero taps.

6. CONCLUSION

Asymmetric multiview video coding is a scenario for reducing transmission bandwidth and computational complexity. However, to support inter-view prediction, conventional solutions require downsampling of high-resolution pictures. Since memory usage and computational complexity are two vitally important aspects for a decoder design, conventional solutions are unfavorable. This paper presented a low-complexity motion compensation method for asymmetric multiview video coding, wherein the downsampling is not needed and the complexity of motion compensation is decreased compared to the conventional motion compensation. Moreover, there is no additional decoded picture buffer needed to store downsampled reference pictures in the proposed method. Simulation results showed that the coding efficiency of the proposed method is essentially identical compared to the coding efficiency of the conventional methods.

7. REFERENCES

- [1] "Joint Draft 5.0 on Multiview Video Coding," *JVT-Y209*, Shenzhen, China, Oct. 2007.
- [2] A. Vetro, W. Matusik, H. Pfister, J. Xin, "Coding Approaches for End-to-End 3D TV Systems," *Picture Coding Symposium*, 2004.
- [3] Julesz B., *Foundations of Cyclopean Perception*, University of Chicago Press, Chicago, IL, USA, 1971.
- [4] C. Fehn, P. Kauff, S. Cho, H. Kwon, N. Hur, J. Kim, "Asymmetric coding of stereoscopic video for transmission over T-DMB," *Proc. 3DTV-CON 2007*, Kos Island, Greece, May 2007.
- [5] H. Kimata, S. Shimizu, K. Kamikura, Y. Yashima, "Inter-view prediction with downsampled reference pictures," *JVT-W079*, San Jose, CA, USA, Apr. 2007.
- [6] Y. Chen, S. Liu, Y.-K. Wang, M. M. Hannuksela, H. Li, "Low complexity asymmetric multiview video coding," *JVT-Y054*, Shenzhen, China, Oct. 2007.
- [7] G. Faria, J. A. Henriksson, E. Stare, and P. Talmola, "DVB-H: Digital Broadcast Services to Handheld Devices," *Proceedings of the IEEE*, vol. 94, no. 1, pp. 194-209, Jan. 2006.
- [8] "Common Test Conditions for Multiview Video Coding," *JVT-T207*, Klagenfurt, Austria, Jul. 2006.
- [9] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *VCEG-M33*, Mar. 2001.
- [10] V. Lappalainen, A. Hallapuro, T.D. Hamalainen, "Complexity of optimized H.26L video decoder implementation," *IEEE Trans. on CSVT*, vol. 13, iss. 7, pp. 717-725, 2003.